Ruby 1.9.3 Enumerable module quick reference

Basic Methods	Notes	Effect	rubies.min
to_a entries		Returns array of all elements.	<= 1.3

count
count(value)
count { |x| … }
		Returns the total count of elements, or the count of elements equal to *value*,
		or the number of elements where the block returns “true”.
			1.8.7

Iteration			

each { |x| … }
each_entry { |x| … }
	
	* ∞
		Yields all elements.	<= 1.3

each_with_index(*args) { |x, i| … }	* ∞	Yields all elements with their indices. *args* are passed through to #each.	<= 1.3
each_cons(n) { |x| … }	* ∞	Yields each possible array of *n* consecutive elements.	1.8.7
each_slice(n) { |x| … }	* ∞	Yields disjoint slices of *n* consecutive elements.	1.8.7
cycle(times=nil) { |x| … }	*	Yields all elements repeatedly forever or for specified number of times.	1.8.7
reverse_each { |x| … }	*	Yields all elements in reverse order.	1.8.7

Questions			
include?(value) member?(value)		Returns true if any element == *value*.	<= 1.3
all? [{|x| … }]		Returns true if block never returns “false”. Default block { |x| x }.	1.8.7
any? [{|x| … }]		Returns true if block ever returns “true”. Default block { |x| x }.	1.8.7
none? [{ |x| … }]		Returns true if block never returns “true”. Default block { |x| x }.	1.8.7
one? [{ |x| … }]		Returns true if block returns “true” exactly once. Default block { |x| x }.	1.8.7

Sorting			
sort [{ |a, b| … }]		Returns array sorted by <=> operator or by block. Default block { |a, b| a <=> b }.	<= 1.3
sort_by { |x| … }	*	Returns array sorted by the return value of the block.	1.8.5
max [{ |a, b| … }]		Returns maximum element. Default block { |a, b| a <=> b }.	<= 1.3
max_by { |x| … }	*	Returns element with maximum block return value.	1.8.7
min [{ |a, b| … }]		Returns minimum element. Default block { |a, b| a <=> b }.	<= 1.3
min_by { |x| … }	*	Returns element with minimum block return value.	1.8.7
minmax [{ |a, b| … }]		Returns [min, max]. Default block { |a, b| a <=> b }.	1.8.7
minmax_by { |x| … }	*	Returns [min, max] using block return value.	1.8.7

Searching for one element			

detect(ifnone = nil) { |x| … }
find(ifnone=nil) { |x| … }
	* ∞	Returns first element where block returns “true”.	<= 1.3

find_index(value=nil)
find_index { |x| … }
	* ∞	Returns index of the first element where block returns “true”.	1.8.7

Filtering by value			

find_all { |x| … }
select { |x| … }
		Returns array of all elements where block returns “true”.	<= 1.3

reject { |x| … }		Returns array of all elements where block returns “false”.	<= 1.3
grep(pattern) [{ |x| … }]		Returns array of block return values for elements where *pattern* === *element*. Default block { |x| x }.	<= 1.3

Filtering by position in series			
first first(n)	∞	Returns first element or array of first *n* elements.	1.8.7
take(n)	∞	Returns array of first *n* elements.	1.8.7
take_while { |x| … }	∞	Returns array of all elements before the first one where the block returned “false”.	1.8.7
drop(n)		Returns array of all elements after first *n*.	1.8.7
drop_while { |x| … }		Returns array of all elements starting with the one where the block returned “false”.	1.8.7

Dividing into subsets			

chunk { |x| … }
chunk(obj) { |x, obj| … }
	∞	Returns enumerator for consecutive chunks of elements with common block value.
		Special effect if block returns nil, :_separator, or :_alone.
			1.9.2

slice_before(pattern)
slice_before { |x| … }
slice_before(obj) { |x, obj| … }
	∞	Returns enumerator for consecutive chunks of elements. If *pattern* === *element* or
		block returns true, that element is the beginning of a chunk.
			1.9.2

partition { |x| … }		Returns [true_array, false_array].	1.8.5
group_by { |x| … }		Returns a hash associating block return values to arrays of elements.	1.8.7

Other			

collect { |x| … }
map { |x| … }
	*	Returns an array with the results from block.	<= 1.3

collect_concat { |x| … }
flat_map { |x| … }
	*	Returns a new array made by concatenating all block results.
		Usually equivalent to map { |x| … }.flatten
			1.9.2

inject(initial, sym)
inject(sym)
inject(initial) { |memo, x| … }
inject { |memo, x| … }
reduce ...
		Combines all elements by applying binary operation specified by block or symbol.
		memo is the last return value from the block.
		Returns the last return value from the block.
			1.8.5

each_with_object(o) {|x, o| … }	*	Iterates and then returns *o*. Similar to #inject but block return value is ignored.	**1.9.1**
zip(*enums) → array_of_arrays zip(*enums) { |arr| ... } → nil	∞	Merges enum with other enums to make arrays that each have an element from each enum; same length as original enum.	1.8.5

* - If no block is passed, an enumerator is returned and the normal function is deferred.
∞ - Can be used with infinite series (supports lazy evaluation)

2012-03-20
davidegrayson.com